UCRL-CR-151855

Distributed Data Integration Infrastructure

Terence Crithlow, Bertram Ludaescher, Mladen Vouk, Calton Pu

February 24, 2003

DISCLAIMER

This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

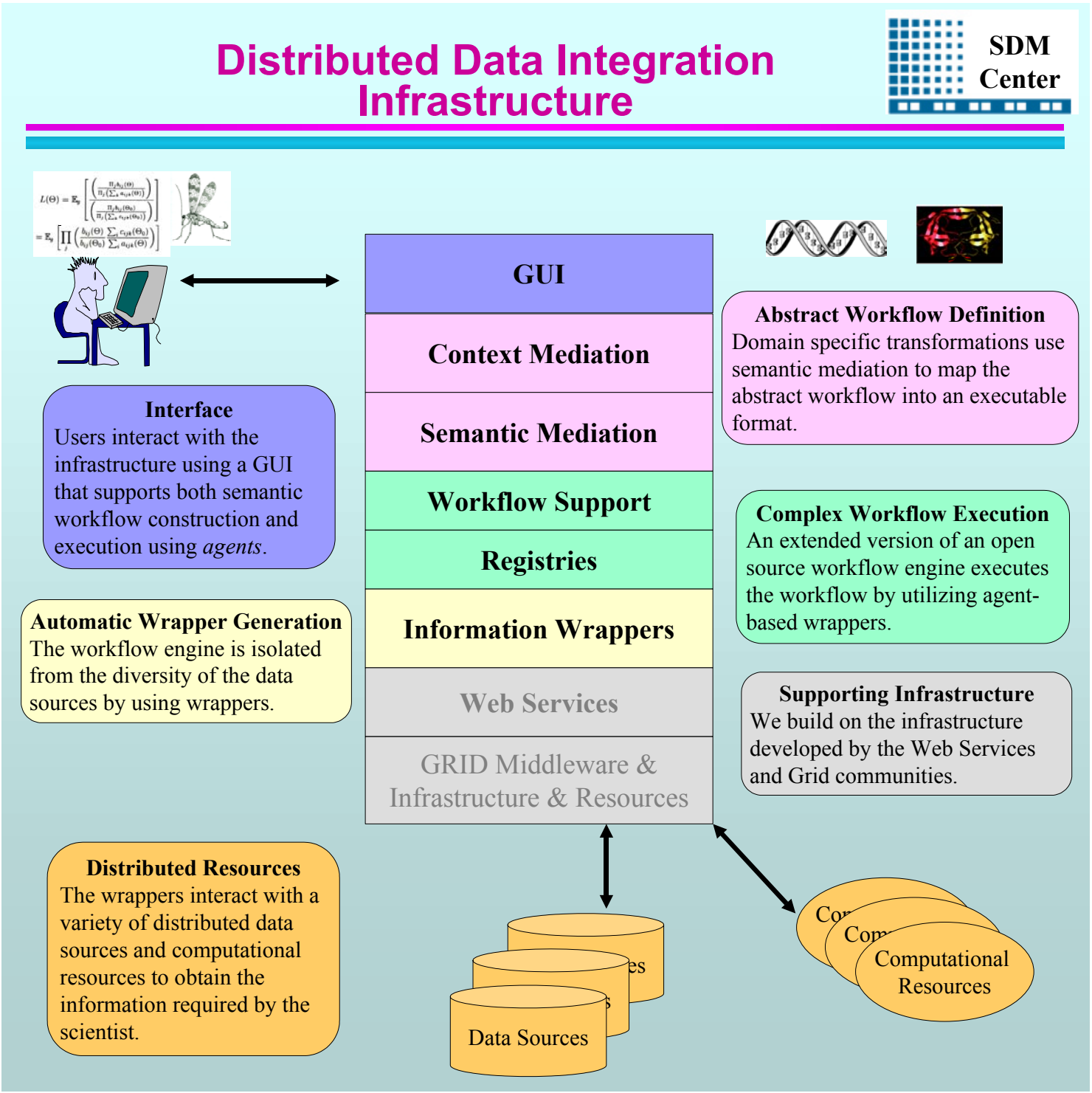

Figure 1 Data Integration Infrastructure

Area 1: Distributed, Heterogeneous Data Integration (Biology)

Introduction

The Internet is becoming the preferred method for disseminating scientific data from a variety of disciplines. This can result in information overload on the part of the scientists, who are unable to query all of the relevant sources, even if they knew where to find them, what they contained, how to interact with them, and how to interpret the results. A related issue is keeping up with current trends in information technology often taxes the end-user's expertise and time. Thus instead of benefiting from this information rich environment, scientists become experts on a small number of sources and technologies, use them almost exclusively, and develop a resistance to innovations that can enhance their productivity. Enabling information based scientific advances, in domains such as functional genomics, requires fully utilizing all available information and the latest technologies.

In order to address this problem we are developing a end-user centric, domain-sensitive workflow-based infrastructure, shown in Figure 1, that will allow scientists to design complex scientific workflows that reflect the data manipulation required to perform their research without an undue burden. We are taking a three-tiered approach to designing this infrastructure utilizing 1) abstract workflow definition, construction, and automatic deployment, 2) complex agent-based workflow execution and 3) automatic wrapper generation. In order to construct a workflow, the scientist defines an abstract workflow (AWF) in terminology (semantics and context) that is familiar to him/her. This AWF includes all of the data transformations, selections, and analyses required by the scientist, but does not necessarily specify particular data sources. This abstract workflow is then compiled into an executable workflow (EWF, in our case XPDL) that is then evaluated and executed by the workflow engine. This EWF contains references to specific data source and interfaces capable of performing the desired actions. In order to provide access to the largest number of resources possible, our lowest level utilizes automatic wrapper generation techniques to create information and data wrappers capable of interacting with the complex interfaces typical in scientific analysis. The remainder of this document outlines our work in these three areas, the impact our work has made, and our plans for the future.

Abstract Workflow Definition

A scientist describes his/her workflow in abstract, but semantically familiar terms and in a familiar context. For example, an AWF can be a directed graph, in which task nodes represent abstract (or virtual tasks). These abstract tasks do not have to deal with low-level intricacies of the existing web services that actually perform the tasks.

In order to map the abstract tasks to executable web services, an abstract-as-view (AAV) definition has to be provided. For example, the abstract task

"promoters", which computes the promoters of a gene, involves several smaller steps including sequence lookups at Genbank and BLAST computations. We describe AAV mappings in a logical rule language. The scientist/end-user will in general not need to deal with these definitions but only with the abstract tasks exported by them.

During the AWF design phase, when abstract tasks are chained together, semantics of task inputs and outputs are used to constrain to type-correct AWFs. Similarly, when the AWF has been translated into an executable workflow (EWF) of either local or web service invocations, the EWF is checked for data type correctness – if necessary, conversion steps are inserted (e.g., in the case of a genetic sequence complementation of 5' to 3' sequences or vice versa).

To date, the semantic and context part of the system is implemented as a "hard-wired" prototype as a CGI-based Perl tool. It implements two variants of "Matt's Promoter Identification Workflow" (PIW), shown in Figure 2. The lessons learnt, in particular about the details of the application domain and PIW workflow, have been incorporated into the design of the generic service-based workflow a management prototype being developed by the P1 team and demonstrated at SC'02.

The main achievement was the development of a revised system architecture in which a distinction is made between "scientist friendly" abstract workflows and executable workflows of web services. An initial approach to the technical challenges

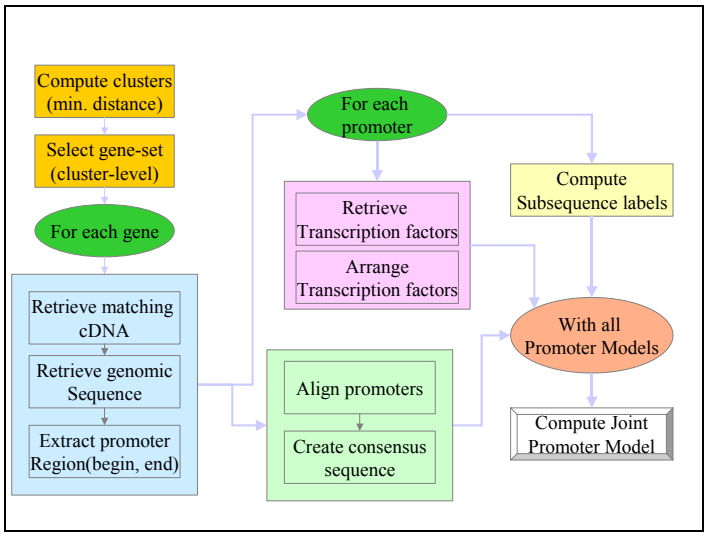

Figure 2 Matt's Promoter Identification Workflow

when translating abstract to executable workflows has been devised: the unfolding of a subset of AWFs (e.g., without recursion) into EWFs using AAV mappings. Currently, we use XPDL as the target EWF language.

Once the automatic AWF->EWF translation has been implemented and integrated into the prototype, we will be able to study the difference in effort between designing AWFs vs. EWFs. For now, there still remain several unresolved issues that make the translation technically challenging (e.g., the unfolding of recursive AAVs).

Complex Workflow Execution

We have developed and deployed an Alpha-prototype of the workflow construction and execution infrastructure and agents. To do that we are leveraging the workflow analysis and profiling effort described in the previous section . While the work is intended to result in a technology applicable to service agent-based support of a general scientific workflow, current research and development activities are concentrating around bioinformatics workflows. Support of a workflow is envisioned as a

suite of services that are being delivered over the network using appliance-like interfaces, either web-based (through browsers) or specially developed but portable interfaces (e.g., either Java-based programs running on the local platform, or web-browser interfaces). A set of primitive, domain-specific services have been developed and the solution will be generalized once they are well understood. We currently have basic service discovery, composition, data-access and data-analysis agent-based workflow support systems in place. Semantic and context analysis is still manual and being studied.

Specifically, we have

a) An integrated service-based support system for "Matt's" workflows in alpha production state.
b) A GUI service composition, recording and playback sub-system based on open-source workflow engines and editors (Ofbiz workflow engine and JProcessEditor). This combination is creates and executes XPDL-based Workflows. The latest version of the GUI prototype is a browser-based applet, and supports complex workflows. The older version is a Java-based locally installed agent.
c) UDDI service registration server and toolset (IBM UDDI)
d) Services that are being delivered using Apache/Tomcat/AXIS-based SOAP.

Automatic Wrapper Generation

To effect automatic canonical information wrapping, and thus insulate workflow construction process from low-level syntactic diversity among heterogeneous data and service sources, we are currently pursuing three related research efforts: automatic wrapper generation, next generation web page clustering, and service selection enabled by adaptive query routing.

XWRAP Composer generates wrapper code capable of recursively extracting information from multiple web/service pages instead of a single page. Most existing wrapper technology is only capable of extracting information from a single Web document. In this domain, however, a single query requires accessing multiple pages with different data structures. The main challenges to wrapping multiple pages are the need to capture the query control flow and the need to encode domain-specific semantic relationships between the pages. These wrappers generated by our Composer will be incorporated into the workflow engine.

We have designed and developed an interface language and Scripting Language as components of the XWRAPComposer toolkit. The interface language allows wrapper developers to specify the interface used to invoke wrappers and the output format of the wrappers, including the object type, structure, and encoding scheme (e.g., XML, plain text, HTML, etc.). The scripting language provides wrapper developers a mechanism to encode control logic and extraction flow of multiple pages into the wrapper generation process.

So far, we have produced a number of XWRAPComposer wrappers and their associated scripts, which have been used in our case-study pilot Bioinformatics scenarios. We plan to deliver the first release of the XWRAP Composer system by the end of March. Three

XWRAP papers have appeared since fall 2001.

We continue to enhance the XWRAPComposer design and development and plan to incorporate a WSDL specification and SOAP interface to each wrapper generated by the Composer. In addition, we will add caching capability and continuous monitoring function into the wrapper code generated, allowing the wrappers to display the original pages where the information was extracted and to show the steps it took to obtain the content extracted.

We are working on new web-page clustering techniques that can outperform the current XWRAP heuristics. This new approach, THOR, is a two-phase clustering system that combines data clustering techniques with IR vector model to identify object-rich content regions in Web pages. The first phase clusters pages to discover answer pages with different templates, allowing the separation of content-rich answer pages from error pages and exception pages. The second phase clustering identifies all content rich regions in the answer pages. We are currently working on the initial implementation of THOR.

Service selection enabled by adaptive query routing is required to select the best source to answer a query. If several sites contain effectively the same information, selecting the best site based on current performance is a challenging task, yet may dramatically impact the usability of a system. We are investigating approaches for dynamically selecting the best site based on current resource distributions.

We are planning to complete the design and development of the first prototype of THOR in Summer 2003 and then incorporate it into the XWRAP Elite, an existing XWRAP system to test its ability.

We have built an adaptive query routing system that can route queries to appropriate information sources based on the source capability profiles. We encode source query capability into the source profiles and combine with user profiles to create matching between users' queries and source profiles. We are currently studying other alternative mechanisms to route queries, including using document-term frequency information as a means to model source capability.

We plan to complete the design and development of the first prototype of the adaptive query routing (AQR) enabled Service selection system by end of 2003.

Impact on Applications

Our domain scientist (Matt Coleman) is now using the prototypes and "custom workflows" developed by this effort. This prototype implements an in part hard-wired workflow that is directly relevant to Matt's research goals. This prototype utilizes wrappers generated by XWRAPComposer, the end-user GUI developed by our team, and services hosted on project servers. While we have duplicated this workflow using our initial, distributed workflow infrastructure, we have not yet upgraded Matt to this more flexible workflow environment.

Despite having access to only our initial prototype, Matt has used our

infrastructure to perform research that has lead to two scientific papers in significantly less time than would have been possible using traditional approaches. This early success indicates that our three-tiered approach to developing a workflow-based infrastructure for scientific data integration is a promising approach.

Publications

[ABB03] I. Altintas, S. Bhagwanani, D. Buttler, S. Chandra, Z. Cheng, M. Coleman, T. Critchlow, A. Gupta, W. Han, L. Liu, B. Ludaescher, C. Pu, R. Moore, A. Shoshani, M. Vouk, A Modeling and Execution Environment for Distributed Scientific Workflows, submitted for publication, 2003, (system demonstration paper).

[Alt02] Ilkay Altintas, On the Generation of Workflows over Heterogeneous Sources and Services, EDBT 2002 Summer School: DBJunior, (presentation).

[BCC02] David Buttler, Matthew Coleman, Terence Critchlow, Renato Fileto, Wei Han, Calton Pu, Daniel Rocco, Li Xiong. Querying Multiple Bioinformatics Information Sources: Can Semantic Web Research Help? SIGMOD Record, Vol 31, No. 4, December 2002.

[CBL03] James Buchanan Caverlee, David Buttler, and Ling Liu. ``Discovering Objects in Dynamically-Generated Web Pages '', Submitted.

[Che02] Zhengang Cheng, "Incorporating Agent Behavior into Web Services," presented at the 40th ACMSE Conference, held in Raleigh, NC, 26-27 April, 2002

[CSV02] Zhengang Cheng, Munindar P. Singh, and Mladen A. Vouk, "Composition Constraints for Semantic Web Services," presented at the WWW-2002 workshop on "Real World RDF and Semantic Web Applications"

[CSV02b] Zhengang Cheng, Munindar P. Singh, and Mladen A. Vouk, "Composition Constraints for Semantic Web Services," accepted for publication as a chapter in the book "Real World Semantic Web Applications", IOS Press, editor V. Kashyap, 2002

[HBP01] Wei Han, David Buttler, Calton Pu. Wrapping Web Data into XML SIGMOD Record, July 2001

[LAG02] B. Ludaescher, I. Altintas, A. Gupta, M. Martone, X. Qian, Data Integration and Mediation, NPACI All Hands Meeting, San Diego, 2002, (tutorial).

[LAG02b] B. Ludaescher, I. Altintas, A. Gupta, Time to Leave the Trees: From Syntactic to Conceptual Querying of XML, Intl. Workshop on XML Data Management, in conjunction with EDBT, Prague, March 2002, LNCS 2490.

[LAG02c] B. Ludaescher, I. Altintas, A. Gupta, Proposal for an Executable Workflow (EWF) Language, 2002, SciDAC/SDSC Tech Report.

[LAG02d] B. Ludaescher, I. Altintas, A. Gupta, A Semantic Mediation Approach for Scientific Workflows, 2002, SciDAC/SDSC Tech Report

[LAG03] B. Ludaescher, I. Altintas, A. Gupta, Compiling Abstract Scientific Workflows into Web Service Workflows, submitted for publication, 2003.

[LBC03] Ling Liu, David Buttler, Terence Critchlow, Wei Han, Henrique Paques, Calton Pu, Dan Rocco. "BioZoom: Exploiting Source-capability

Information for Integrated Access to Multiple Bioinformatics Data Sources" To appear in the Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering (BIBE 2003), 10-12 March 2003, Washington, DC.

[LPH01] Ling Liu, Calton Pu, Wei Han. "An XML-Enabled Data Extraction Tool for Web Sources" *International Journal of Information Systems*, Special Issue on Data Extraction, Cleaning, and Reconciliation. (eds. Mokrane Bouzeghoub and Maurizio Lenzerini), 2001.

[PLP02] Henrique Paques, Ling Liu, Calton Pu. ``Ginga: A Self-Adaptive Query Processing Service". Proceedings of ACM International Conference on Information and Knowledge Management (CIKM 2002).

[PLP03] Henrique Paques, Ling Liu, and Calton Pu. Distributed Query Adaptation and Its Trade-offs. To appear in Proceedings of the Eighteenth Annual ACM Symposium on Applied Computing (ACM SAC 2003) , March 9 to 12, 2003, Melbourne, Florida, USA. (DataBase Systems track)

Software:

http://sdm.ncsu.edu (userid: sdm passwd: *sdm!)

Applet of the JProcessEditor (http://sdm2.csc.ncsu.edu:8080/applet/applet.html)

Appendix A: People currently involved in the project

LLNL
Terence Critchlow (CASC)
Matt Coleman (BBRP)

Georgia Tech
Faculty: Ling Liu, Calton Pu
Students: David Buttler, Wei Han, Henrique Paques

NCSU
Faculty: Mladen A. Vouk, Donald L. Bitzer, Munindar Singh,
Post-doc: David Rosnick
Students: Sandeep Chandra, Zhengang Cheng, Dan Shupp, S. Bhagwanani

SDSC
Faculty: Bertram Ludaescher, Amarnath Gupta
Staff: Ilkay Altintas